

PYTHON BASED SPEAKING MEDICATION REMAINDER

Submitted in partial fulfillment of the requirements for the award of Bachelor
of Engineering Degree in Electronics and Communication Engineering

by

SEERAM UPENDRA (38130199)

SATHI JAGAN RAM MOHAN REDDY(38130197)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SEERAM UPENDRA** (38130199) and **SATHI JAGAN RAM MOHAN REDDY**(38130197) who carried out the project entitled "**PYTHON BASED SPEAKING MEDICATION REMAINDER**" under our supervision from December 2020 to March 2021.

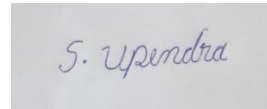
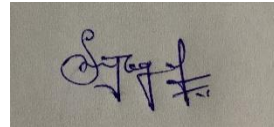
Internal Guide
(Dr.S.LAKSHMI)

Head of the Department
(Dr. T. RAVI, M.E., Ph.D.,)

Submitted for Viva voce Examination held on _____
Internal Examiner External Examiner

DECLARATION

We **SEERAM UPENDRA** (38130199) and **SATHI JAGAN RAM MOHAN REDDY** (38130197) hereby declare that the Project Report entitled “**PYTHON BASED SPEAKING MEDICATION REMAINDER PROJECT** ” done by us under the guidance of **Dr. S.LAKSHMI** at Sathyabama Institute of Science and Technology , Chennai – 600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Electronics And Communication Engineering.



DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITA, Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr.S.LAKSHMI** for her valuable guidance, suggestions and constant encouragement during our project.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Most of times patients may forget to take the medicines at proper time as per their prescription which may cause in late recovery from the disease/illness. In case of some specific diseases like diabetes, hypertension, anemia , medication mix-ups or improper dosages prove dangerous to patients. So, it is necessary to take proper medicines in proper quantity at proper time. Medication reminder solutions have become a matter of concern in health-care research, as advancements in treatments and medication may be offset by patients' low attention to medication schedule. There is need for a strong automatic system to remind patient about the concerned medication which will assist them for the medication and one can avoid the adverse effects of improper medication. Here I am presenting “PYTHON BASED SPEAKING MEDICATION REMINDER ” which will fulfill the concerned purpose. This project will help elder and illiterate people with their regular medication by reminding them time to time.

Chapter	Title	Page No
1	Introduction	1
2	Literature Survey	2
3		
3.1	Aim	3
3.2	Existing System	3
3.3	Proposed System	3
3.4	Block Diagram	3
4		
4.1	Features of the System	4
4.2	Embedded systems	4
4.2.1	Overview of Embedded Systems	4
4.2.2	Function of Embedded Systems	5
4.2.3	Block Diagram of Embedded Systems	6
4.2.4	Characteristics	7
4.2.5	Power Supply	10
4.2.6	Transformers	10
4.2.7	Rectifiers	11
4.2.8	Filters	11
4.2.9	Registers	11
4.2.10	Regulators	12
4.3	IOT	13
4.3.1	Features of IOT	14
4.3.2	Advantages and Disadvantages	15
4.3.3	Embedded Systems in IOT	16

	4.3.4	Embedded System Hardware	17
	4.3.5	IOT Ecosystem	18
	4.3.6	IOT Decision Frame Work	20
	4.3.7	IOT Architechture	22
	4.3.8	Stages of IOT	22
	4.3.9	IOT Devices	25
	4.3.10	Properties of IOT	30
	4.4	RASPBERRY PI	
	4.4.1	Raspberry pi basic Hardware setup	36
	4.4.2	Operating System SD Card	38
	4.4.3	Keyboard and Mouse	38
	4.4.4	Display	39
	4.4.5	Raspberry Pi Port Specification	40
	4.4.6	Power Supply	40
	4.4.7	Cables	40
	4.4.8	Addition Peripherals	41
	4.5	Raspberry Pi Advance Setup	
	4.5.1	Find Hardware and Setting up	42
	4.5.2	Serial Connections	42
	4.5.3	Serial Parameters	42
	4.6	SD Card Setup	44
	4.7	LCD Display	44
	4.8	Functional Description of Controller	46
	4.9	Introduction- Make Your Raspberry Pi Speak	48
5			
	5.1	Discussion	54
	5.2	Hardware Kit	54
	5.3	Result	55
6			
	6.1	Conclusions	56
	6.2	References	57

CHAPTER 1

INTRODUCTION

Need for the Device Any living being which is undergoing through medication can come across life -threatening adverse effects of improper medication habits. These living beings can include human, pets, animals etc. The patients under human being category may include businessman, social workers, politicians, teachers, students, IT employees etc. These people may be busy in their daily routine life schedule or else we can consider elder people who usually forgot to take their medicines. If they are suffering from any kind of serious disease/illness like diabetes, heart problem or some neural disease then it's their duty to take proper medicines in proper quantity at specific intervals as suggested by doctors. Any kind of miss in medicine dosages will directly put their life in threat. This comes to think me out for such medication reminder.

Sources Causing Improper Medication There are sources which causes improper medication:

- a) Irregular medicine intakes due to the patient's busy schedule
- b) Complicated intake schedules due to the large number of medicines taken by the patient
- c) Adverse drug reactions caused by un-reconciled prescriptions obtained from different sources
- d) Lack of knowledge about proper use of medicines
- e) High Dosage Intake of some medicine Memory Loss in Alzheimer Disease

Proposed Solution To solve the problem of medication mix-ups in case of senior citizens and illiterate people suffering from severe diseases, I proposed "Python Based Speaking Medication Reminder " which will remind patient/user for him/her medication.

CHAPTER 2

LITERATURE SURVEY

Miss. Sonali subhash alaspure¹, dr. A.P. Thakare² Mar-2018 [1] the literature reveals a device which will show the medication in take at a particular time. Medication reminder solutions have become a matter of concern in health-care research, as advancements in treatments and medication may be offset by patients' low attention to medication schedule. Medication mix-ups are extremely dangerous. Most of times patients may forget to take the medicines at proper time as per the specification in the prescription which may cause in late recovery from the disease/illness or some life-threatening mistakes

Aswin kumer s v, p.Kanakaraj, I.S.P sairam nadipalli, punya teja a, harini sree t, tejaswni t[2] created a device itself converts the input text into voice commands and speaks out loud so that the patient can hear and take the medication in proper timing and in proper amount. And this device is also integrated with a web cam which can be accessed through anywhere so we can also track the patient current position and status by viewing him from our device. The whole set up is connected to wi-fi and iot data base so we can store every data in cloud so we can access the data of particular medication of every patient which is useful for surveys and improvements

Rasberry PI Medicine Reminder E-Mail Alert Using IOT

The pills to be consumed are popped out of the pillbox at that time. Smart pill box can reduce the guardian's responsibility of giving the correct medicines at the prescribed time. T. It first allows users to enter reminder inputs. System takes input through keyboard to accept various reminders with date and time and dosage. It then reminds patients to take the right medication at the right time. System allows users to store their medication dates and time using raspberry pi. Also users are allowed to enter dosage of each reminder.

CHAPTER 3

3.1 AIM :

The Main Aim of our project is to avoid the mix up of medications and The medicine should be taken on time by remaining the patient time to time.

3.2 EXISTING SYSTEM:

In existing system GSM and Arduino is used for remanding the patient medicine. There is no voice. It will send a message to patient.

3.3 PROPOSED SYSTEM:

To solve the problem of medication mix-ups in case of senior citizens and illiterate people suffering from severe diseases, we proposed “Python Based Speaking Medication Reminder System” which will remind patient/user for him/her medication.

3.4 BLOCK DIAGRAM:

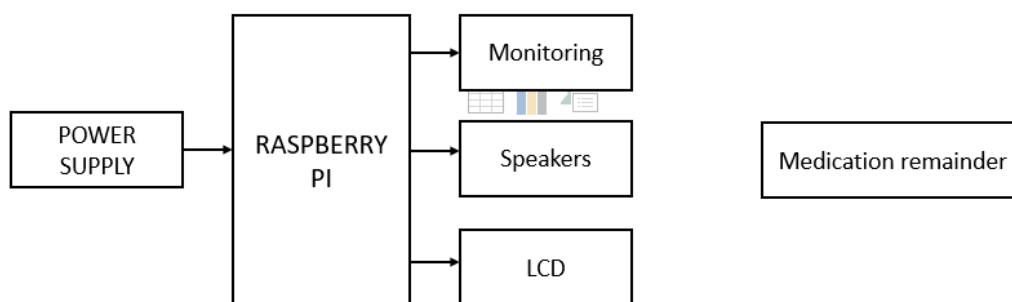


Fig 3.1 Block Diagram

CHAPTER 4

4.1 FEATURES OF THE SYSTEM

- 1] It can be used by elderly people who forgot to take their regular medication.
- 2] It can be easily handled by illiterate users to store inputs according to their convenience.
- 3] This system can save human beings from extreme effect of improper medication mix-ups.
- 4] To assist ADR's affected patients and save them from life threatening events.

4.2 EMBEDDED SYSTEMS

4.2.1 OVERVIEW OF EMBEDDED SYSTEMS

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers

share some elements with embedded systems such as the operating systems and microprocessors which power them but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

4.2.2 FUNCTION BY EMBEDDED SYSTEMS

- Monitor the environment; embedded systems read data from input sensors. This data is then processed and the results displayed in some format to a user or users
- Control the environment; embedded systems generate and transmit commands for actuators.
 - Transform the information; embedded systems transform the data collected in some meaningful way, such as data compression/decompression
 - Although interaction with the external world via sensors and actuators is an important aspect of embedded systems, these systems also provide functionality specific to their applications. Embedded systems typically execute applications such as control laws, finite state machines, and signal processing algorithms. These systems must also detect and react to faults in both the internal computing environment as well as the surrounding electromechanical systems.

There are many categories of embedded systems, from communication devices to home appliances to control systems. Examples include;

- Communication devices
 - e. g.: modems, cellular phones
- Home Appliances
 - e. g.: CD player, VCR, microwave oven
- Control Systems
 - g.: Automobile anti-lock braking systems, robotics, satellite control

4.2.3 BLOCK DIAGRAM OF AN EMBEDDED SYSTEM:

An embedded system usually contains an embedded processor. Many appliances that have a digital interface -- microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an operating system. Others are very specialized resulting in the entire logic being implemented as a single program. These systems are embedded into some device for some specific purpose other than to provide general purpose computing . A typical embedded system is shown in Fig 4.1

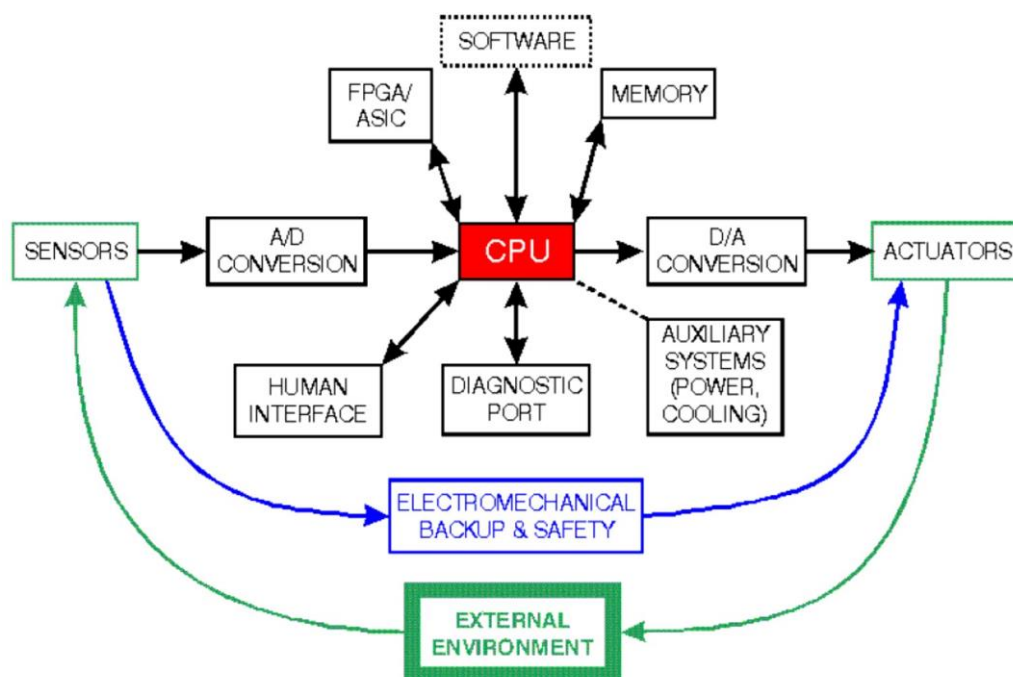


Fig 4.1 Block diagram of a typical embedded system

4.2.4 CHARACTERISTICS OF EMBEDDED SYSTEMS

Embedded systems are characterized by a unique set of characteristics. Each of these characteristics imposed a specific set of design constraints on embedded systems designers. The challenge to designing embedded systems is to conform to the specific set of constraints for the application.

APPLICATION SPECIFIC SYSTEMS:

Embedded systems are not general-purpose computers. Embedded system designs are optimized for a specific application. Many of the job characteristics are known before the hardware is designed. This allows the designer to focus on the specific design constraints of a well-defined application. As such, there is limited user reprogram ability. Some embedded systems, however, require the flexibility of reprogram ability. Programmable DSPs are common for such applications.

REACTIVE SYSTEMS

As mentioned earlier, a typical embedded systems model responds to the environment via sensors and control the environment using actuators. This requires embedded systems to run at the speed of the environment. This characteristic of embedded system is called “reactive”. Reactive computation means that the system (primarily the software component) executes in response to external events. External events can be either periodic or aperiodic. Periodic events make it easier to schedule processing to guarantee performance. Aperiodic events are harder to schedule. The maximum event arrival rate must be estimated in order to accommodate worst case situations. Most embedded systems have a significant reactive component. One of the biggest challenges for embedded system designers is performing an accurate worst case design analysis on systems with statistical performance characteristics (e.g., cache memory on a DSP or other embedded processor). Real time system operation means that the correctness of a computation depends, in part, on the time at which it is delivered. Systems with this requirement must often design to worst case performance. But accurately predicting the worst case may be difficult on complicated architectures. This often leads to overly pessimistic estimates erring on the side of caution. Many embedded systems have a significant requirement for real time operation in order to meet external I/O and control stability requirements. Many real-time systems are also reactive systems.

DISTRIBUTED SYSTEMS

A common characteristic of an embedded system is one that consists of communicating processes executing on several CPUs or ASICs which are connected by communication links. The reason for this is economy. Economical 4 8-bit microcontrollers may be cheaper than a 32-bit processors. Even after adding the cost of the communication links, this approach may be preferable. In this approach, multiple processors are usually required to handle multiple time-critical tasks. Devices under control of embedded systems may also be physically distributed.

HETEROGENEOUS ARCHITECTURES

Embedded systems often are composed of heterogeneous architectures (Fig 4.2). They may contain different processors in the same system solution. They may also be mixed signal systems. The combination of I/O interfaces, local and remote memories, and sensors and actuators makes embedded system design truly unique.

Embedded systems also have tight design constraints, and heterogeneity provides better design flexibility.

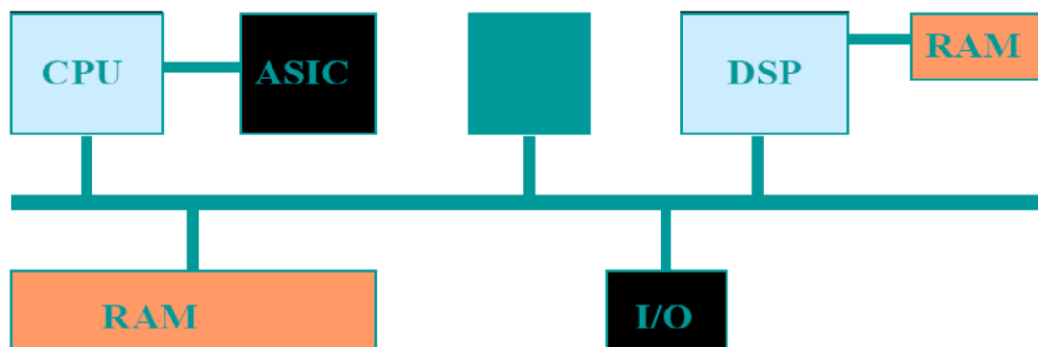


Fig 4.2 Embedded Systems having Heterogeneous Architectures

HARSH ENVIRONMENT

Many embedded systems do not operate in a controlled environment. Excessive heat is often a problem, especially in applications involving combustion (e.g., many transportation applications). Additional problems can be caused for embedded computing by a need for protection from vibration, shock, lightning, power supply fluctuations, water, corrosion, fire, and general physical abuse.

SYSTEM SAFETY AND RELIABILITY

As embedded system complexity and computing power continue to grow, they are starting to control more and more of the safety aspects of the overall system. These safety measures may be in the form of software as well as hardware control. Mechanical safety backups are normally activated when the computer system loses control in order to safely shut down system operation. Software safety and reliability is a bigger issue. Software doesn't normally "break" in the sense of hardware. However software may be so complex that a set of unexpected circumstances can cause software failures leading to unsafe situations. Discussion of this topic is outside the scope of this book, but the challenges for embedded designers include designing reliable software and building cheap, available systems using unreliable components. The main challenge for embedded system designers is to obtain low-cost reliability with minimal redundancy.

CONTROL OF PHYSICAL SYSTEMS

One of the main reasons for embedding a computer is to interact with the environment. This is often done by monitoring and controlling external machinery. Embedded computers transform the analog signals from sensors into digital form for processing. Outputs must be transformed back to analog signal levels. When controlling physical equipment, large current loads may need to be switched in order to operate motors and other actuators. To meet these needs, embedded systems may need large computer circuit boards with many non-digital components. Embedded system designers must carefully balance system tradeoffs among analog components, power, mechanical, network, and digital hardware with corresponding software.

SMALL AND LOW WEIGHT

Many embedded computers are physically located within some larger system. The form factor for the embedded system may be dictated by aesthetics. For example, the form factor for a missile may have to fit inside the nose of the missile. One of the challenges for embedded systems designers is to develop non-rectangular geometries for certain solutions. Weight can also be a critical constraint. Embedded automobile control systems, for example, must be light weight for fuel economy. Portable CD players must be light weight for portability purposes.

COST SENSITIVITY

Cost is an issue in most systems, but the sensitivity to cost changes can vary dramatically in embedded systems. This is mainly due to the effect of computer costs have on profitability and is more a function of the proportion of cost changes compared to the total system cost.

POWER MANAGEMENT

Embedded systems have strict constraints on power. Given the portability requirements of many embedded systems, the need to conserve power is important to maintain battery life as long as possible. Minimization of heat production is another obvious concern for embedded systems.

4.2.5 POWER SUPPLY

All electronic circuits works only in low DC voltage, so we need a power supply unit to provide the appropriate voltage supply for their proper functioning .This unit consists of transformer, rectifier, filter & regulator. AC voltage of typically 230volts rms is connected to a transformer voltage down to the level to the desired ac voltage. A diode rectifier that provides the full wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation . A regulator circuit can use this dc input to provide dc voltage that not only has much less ripple voltage but also remains the same dc value even the dc voltage varies some what, or the load connected to the output dc voltages changes.

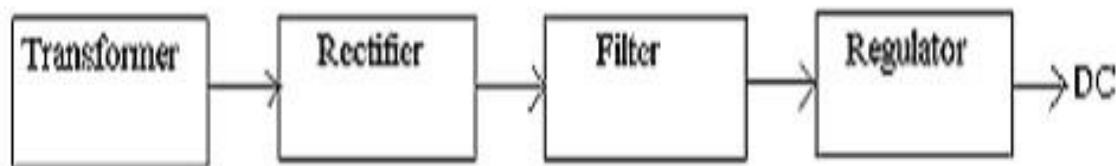


Fig 4.3. General Block Of Power Supply Unit

4.2.6 TRANSFORMER:

A transformer is a static piece of which electric power in one circuit is transformed into electric power of same frequency in another circuit. It can raise or lower the voltage in the circuit, but with a corresponding decrease or increase in current. It works with the principle of mutual induction. In our project we are using a step down transformer to providing a necessary supply for the electronic circuits. Here we step down a 230volts ac into 12volts ac.

4.2.7 RECTIFIER:

A dc level obtained from a sinusoidal input can be improved 100% using a process called full wave rectification. Here in our project for full wave rectification we use bridge rectifier. From the basic bridge configuration we see that two diodes(say D2 & D3) are conducting while the other two diodes (D1 & D4) are in off state during the period $t = 0$ to $T/2$. Accordingly for the negative cycle of the input the conducting diodes are D1 & D4 .Thus the polarity across the load is the same.

In the bridge rectifier the diodes may be of variable types like 1N4001, 1N4003, 1N4004, 1N4005, 1N4007 etc.... can be used. But here we use 1N4007, because it can withstand up to 1000v.

4.2.8 FILTERS:

In order to obtain a dc voltage of 0 Hz, we have to use a low pass filter. So that a capacitive filter circuit is used where a capacitor is connected at the rectifier output & a dc is obtained across it. The filtered waveform is essentially a dc voltage with negligible ripples & it is ultimately fed to the load.

4.2.9 REGISTERS:

The controller IC has two 8 bit registers, an instruction register (IR) and a data register (DR). The IR stores the instruction codes and address information for display data RAM (DD RAM) and character generator RAM (CG RAM). The IR can be written, but not read by the MPU.

The DR temporally stores data to be written to /read from the DD RAM or CG RAM. The data written to DR by the MPU, is automatically written to the DD RAM or CG RAM as an internal operation.

When an address code is written to IR, the data is automatically transferred from the DD RAM or CG RAM to the DR. data transfer between the MPU is then completed when the MPU reads the DR. likewise, for the next MPU read of the DR, data in DD RAM or CG RAM at the address is sent to the DR automatically. Similarly, for the MPU write of the DR, the next DD RAM or CG RAM address is selected for the write operation.

The dot-matrix liquid crystal display controller and driver LSI displays alphanumeric, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver

4.2.10 REGULATORS:

The output voltage from the capacitor is more filtered & finally regulated. The voltage regulator is a device, which maintains the output voltage constant irrespective of the change in supply variations, load variations & temperature changes. Here we use fixed voltage regulator namely LM7805. The IC LM7805 is a +5v regulator which is used for microcontroller

4.2.11 FEATURES AND DESCRIPTION OF REGULATORS

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

4.3 IOT

IoT stands for Internet of Things, which means accessing and controlling daily usable equipments and devices using Internet.

Our IoT tutorial includes all topics of IoT such as introduction, features, advantage and disadvantage, ecosystem, decision framework, architecture and domains, biometric, security camera and door unlock system, devices, etc.

What is an Internet of Things (IoT)

Let's us look closely at our mobile device which contains GPS Tracking, Mobile Gyroscope, Adaptive brightness, Voice detection, Face detection etc. These components have their own individual features, but what about if these all communicate with each other to provide a better environment? For example, the phone brightness is adjusted based on my GPS location or my direction.

Connecting everyday things embedded with electronics, software, and sensors to internet enabling to collect and exchange data without human interaction called as the Internet of Things (IoT).

The term "Things" in the Internet of Things refers to anything and everything in day to day life which is accessed or connected through the internet.

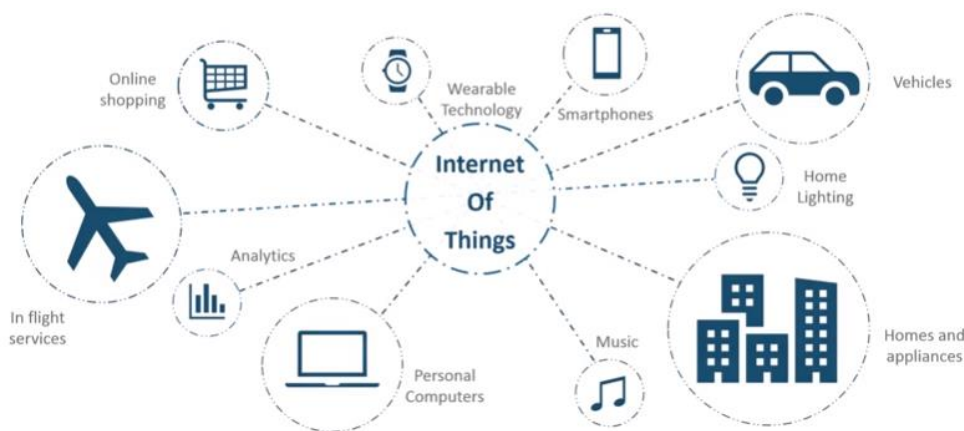


Fig 4.4 Internet Of Things

IoT is an advanced automation and analytics system which deals with artificial intelligence, sensor, networking, electronic, cloud messaging etc. to deliver complete systems for the product or services. The system created by IoT has greater transparency, control, and performance.

As we have a platform such as a cloud that contains all the data through which we connect all the things around us. For example, a house, where we can connect our home appliances such as air conditioner, light, etc. through each

other and all these things are managed at the same platform. Since we have a platform, we can connect our car, track its fuel meter, speed level, and also track the location of the car.



If there is a common platform where all these things can connect to each other would be great because based on my preference, I can set the room temperature. For example, if I love the room temperature to be set at 25 or 26-degree Celsius when I reach back home from my office, then according to my car location, my AC would start before 10 minutes I arrive at home. This can be done through the Internet of Things (IoT).

How does Internet of Thing (IoT) Work?

The working of IoT is different for different IoT echo system (architecture). However, the key concept of there working are similar. The entire working process of IoT starts with the device themselves, such as smartphones, digital watches, electronic appliances, which securely communicate with the IoT platform. The platforms collect and analyze the data from all multiple devices and platforms and transfer the most valuable data with applications to devices.

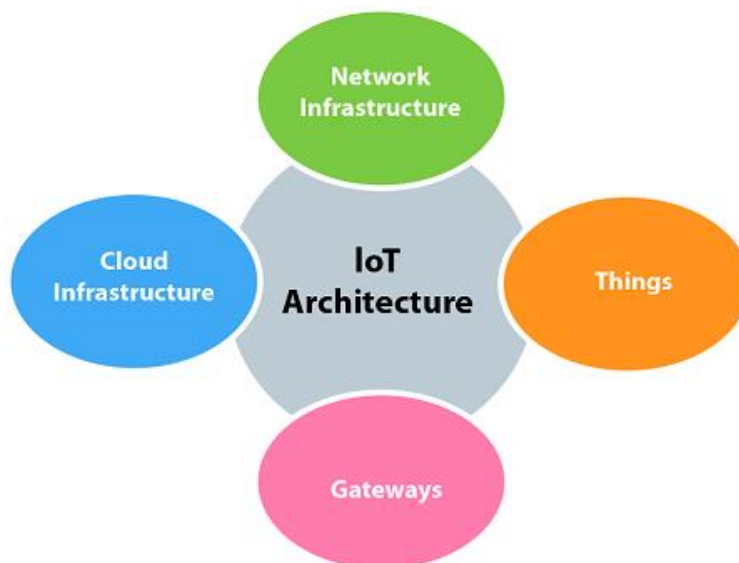


Fig 4.5 IOT Architechture

4.3.1 FEATURES OF IOT

The most important features of IoT on which it works are connectivity, analyzing, integrating, active engagement, and many more. Some of them are listed below:

Connectivity: Connectivity refers to establish a proper connection between all the things of IoT to IoT platform it may be server or cloud. After connecting the IoT devices, it needs a high speed messaging between the devices and cloud to enable reliable, secure and bi-directional communication.

Analyzing: After connecting all the relevant things, it comes to real-time analyzing the data collected and use them to build effective business intelligence. If we have a good insight into data gathered from all these things, then we call our system has a smart system.

Integrating: IoT integrating the various models to improve the user experience as well.

Artificial Intelligence: IoT makes things smart and enhances life through the use of data. For example, if we have a coffee machine whose beans have going to end, then the coffee machine itself order the coffee beans of your choice from the retailer.

Sensing: The sensor devices used in IoT technologies detect and measure any change in the environment and report on their status. IoT technology brings passive networks to active networks. Without sensors, there could not hold an effective or true IoT environment.

Active Engagement: IoT makes the connected technology, product, or services to active engagement between each other.

Endpoint Management: It is important to be the endpoint management of all the IoT system otherwise, it makes the complete failure of the system. For example, if a coffee machine itself order the coffee beans when it goes to end but what happens when it orders the beans from a retailer and we are not present at home for a few days, it leads to the failure of the IoT system. So, there must be a need for endpoint management.

4.3.2 ADVANTAGES AND DISADVANTAGES OF (IOT)

Any technology available today has not reached to its 100 % capability. It always has a gap to go. So, we can say that **Internet of Things** has a significant technology in a world that can help other technologies to reach its accurate and complete 100 % capability as well.

Let's take a look over the major, advantages, and disadvantages of the Internet of Things.

Advantages of IoT

Internet of things facilitates the several advantages in day-to-day life in the business sector. Some of its benefits are given below:

- **Efficient resource utilization:** If we know the functionality and the way that how each device work we definitely increase the efficient resource utilization as well as monitor natural resources.
- **Minimize human effort:** As the devices of IoT interact and communicate with each other and do lot of task for us, then they minimize the human effort.
- **Save time:** As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.
- **Enhance Data Collection:**
- **Improve security:** Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

Disadvantages of IoT

As the Internet of things facilitates a set of benefits, it also creates a significant set of challenges. Some of the IoT challenges are given below:

4.3.3 EMBEDDED DEVICES (SYSTEM) IN (IOT)

It is essential to know about the embedded devices while learning the IoT or building the projects on IoT. The embedded devices are the objects that build the unique computing system. These systems may or may not connect to the Internet.

An embedded device system generally runs as a single application. However, these devices can connect through the internet connection, and able communicate through other network devices.

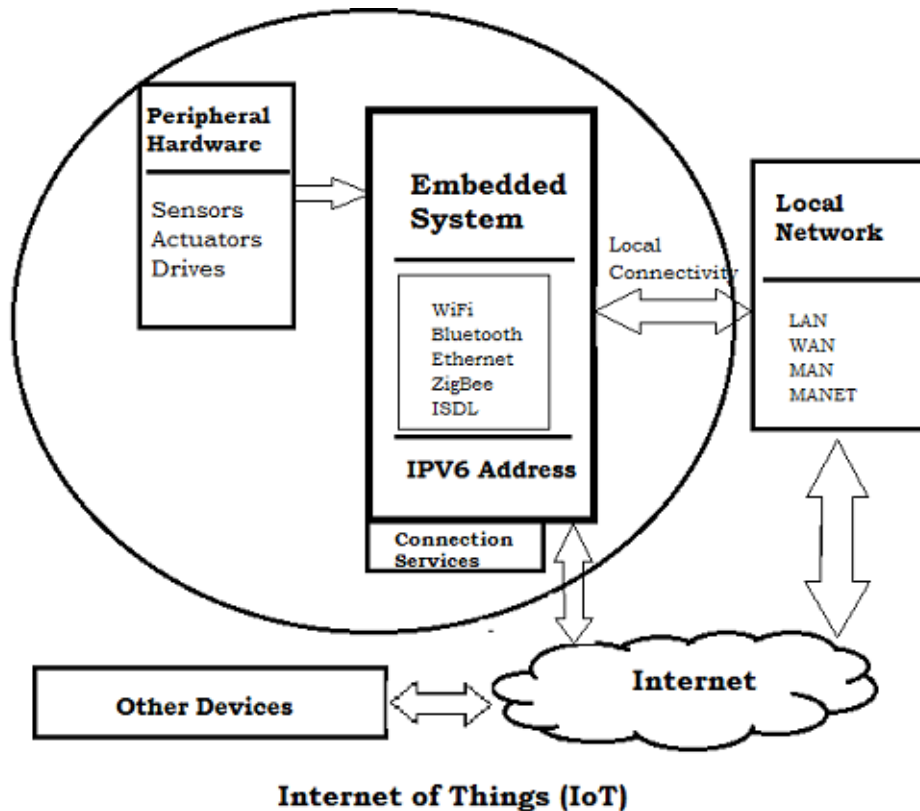


Fig 4.6

4.3.4 EMBEDDED SYSTEM HARDWARE

The embedded system can be of type microcontroller or type microprocessor. Both of these types contain an integrated circuit (IC).

The essential component of the embedded system is a RISC family microcontroller like Motorola 68HC11, PIC 16F84, Atmel 8051 and many more. The most important factor that differentiates these microcontrollers with the microprocessor like 8085 is their internal read and writable memory. The essential embedded device components and system architecture are specified below.

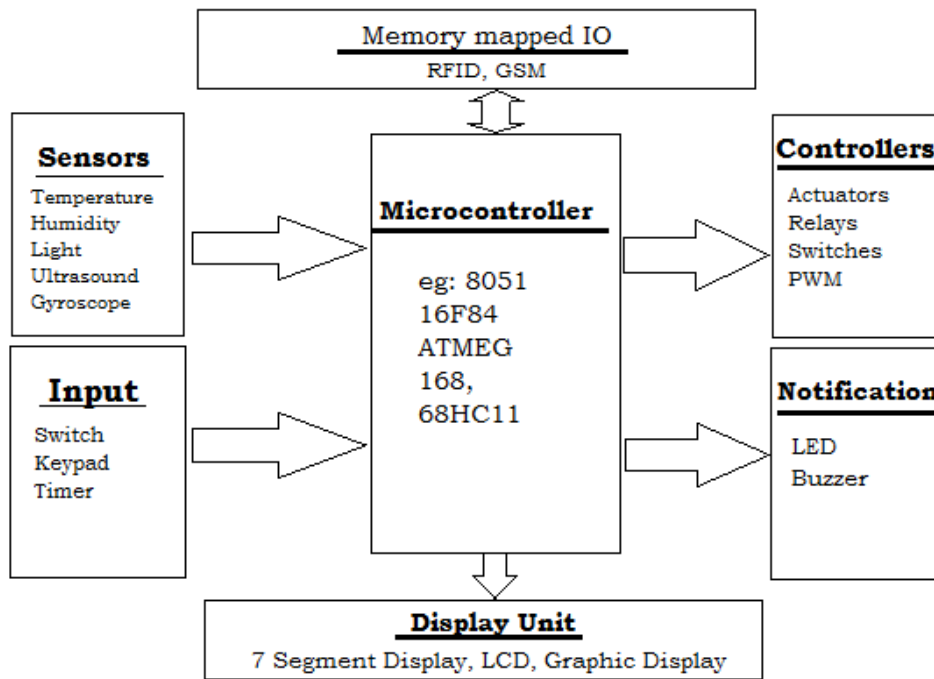


Fig 4.7: Basic Embedded System

EMBEDDED SYSTEM SOFTWARE

The embedded system that uses the devices for the operating system is based on the language platform, mainly where the real-time operation would be performed. Manufacturers build embedded software in electronics, e.g., cars, telephones, modems, appliances, etc. The embedded system software can be as simple as lighting controls running using an 8-bit microcontroller. It can also be complicated software for missiles, process control systems, airplanes etc.

4.3.5 IOT ECOSYSTEM

The IoT ecosystem is not easy to define. It is also difficult to capture its proper image due to the vastness and emerging possibility and the rapidity with which it is expanding in the entire sector. However, the IoT ecosystem is a connection of various kind of devices that sense and analyze the data and communicates with each other over the networks.

In the IoT ecosystem, the user uses smart devices such as smartphones, tablet, sensors, etc. to send the command or request to devices for information over the

networks. The device response and performs the command to send information back to the user through networks after analyzed.

The typical IoT ecosystem is shown in below image, where the smarter devices send and receive data from the devices themselves in the environment that are integrate over network and Cloud Computing.

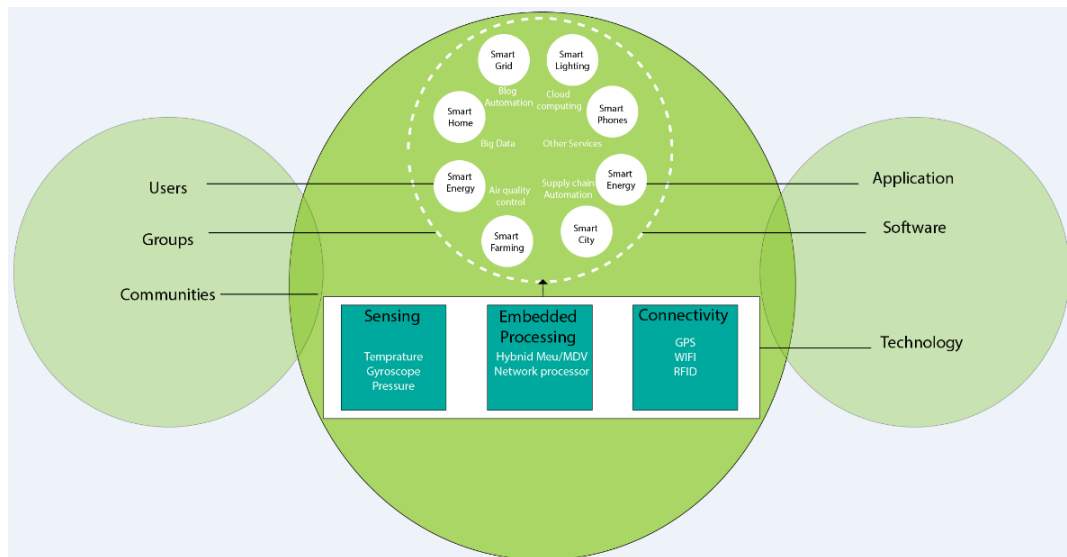


Fig 4.8 Ecosystem IOT

The IoT is itself an ecosystem of network devices that transfer the data. It is also well interconnected with Big Data and Cloud Computing.

- **Sensing, Embedded processing, Connectivity:** The IoT ecosystem senses its surrounding like temperature, gyroscope, pressure, etc. and make the embedded processing using devices. These devices are connected through any type of devices such as GPS, WiFi, RFID, etc. over the networks.
- **Smart devices and environment, Cloud Computing, Big Data:** The data transfer or receive through smart devices and environments are communicated through Cloud Computing or others Servers and stored as Big Data.
- **Technology, Software, Application:** The IoT ecosystem uses any of different technologies, software and application to communicate and connect with smart devices and environment.

- **Users or groups of community:** The product or services generated by the IoT ecosystem are consumed by the users or the group of communities to serve the smart life.

4.3.6 IOT DECISION FRAMEWORK

The IoT decision framework provides a structured approach to create a powerful IoT product strategy. The IoT decision framework is all about the strategic decision making. The IoT Decision Framework helps us to understand the areas where we need to make decisions and ensures consistency across all of our strategic business decision, technical and more.

The IoT decision framework is much more important as the product or services communicates over networks goes through five different layers of complexity of technology.

1. Device Hardware
2. Device Software
3. Communications
4. Cloud Platform
5. Cloud Application



The IoT Technology Stack

Fig 4.9 Technology Stack

DECISION AREA

The IoT decision framework pays attention to six key decision areas in any IoT product. These decision areas are:

1. User Experience (UX)

2. Data
3. Business
4. Technology
5. Security
6. Standards & Regulations

Each of these decision areas is evaluated at each of the IoT Technology Stack. The User Experience will be evaluated at Device Hardware, Device Software and so to provide the better user experience. Then at the next step Data Decision Area, we have to explore data considerations for all the stages of IoT Technology Stack.

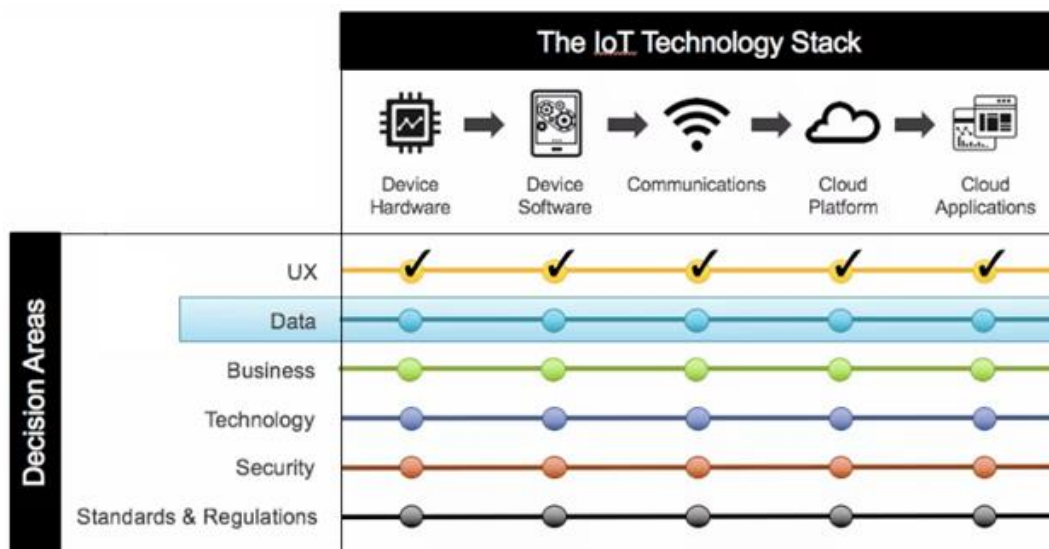


Fig 4.10

DECISION AREA OF THE IOT DECISION FRAMEWORK

Let's see each of the Decision Area of IoT Decision Framework in detail:

1. **User Experience Decision Area:** This is the area where we concentrate about who are the users, what are their requirements and how to provide a great experience at each step of IoT stack without worrying about the technical details.
2. **Data Decision Area:** In this area, we make the overall data strategy such as the data flow over the entire IoT stack to fulfill the user's requirements.

3. **Business Decision Area:** Based on the previous decisions area, we make the decision how product or services will become financial potential. At each of the IoT Stack level are monetized about the costs of providing services.
4. **Technology Decision Area:** In this area, we work with the technology for each layer to facilitate the final solution.
5. **Security Decision Area:** After going through the implementation of technology it is important to decide and provide the security at each stage of the IoT Stack.
6. **Standards & Regulations Decision Area:** At the last stage of IoT Decision Area, we identify the standards and regulations of product or services that will affect your product at each layer of the IoT Stack.

4.3.7 IOT ARCHITECTURE

There is not such a unique or standard consensus on the Internet of Things (IoT) architecture which is universally defined. The IoT architecture differs from their functional area and their solutions. However, the IoT architecture technology mainly consists of four major components:

Components of IoT Architecture

- Sensors/Devices
- Gateways and Networks
- Cloud/Management Service Layer
- Application Layer

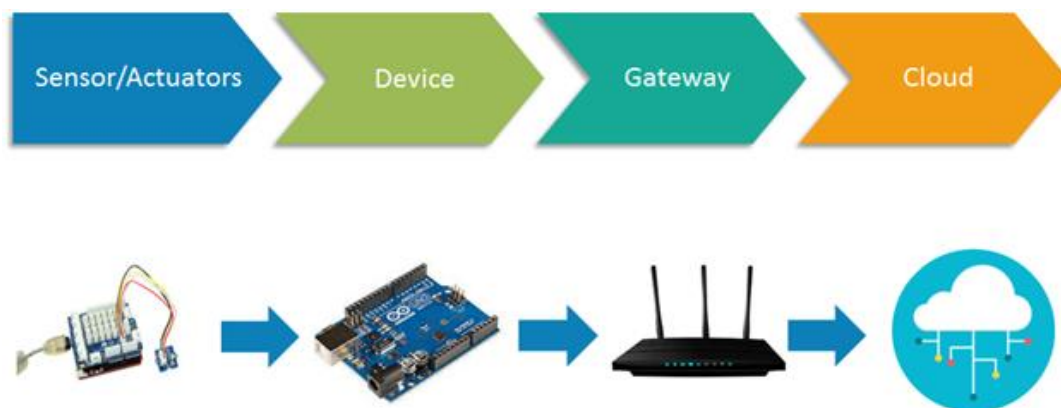


Fig 4.11 IOT Architecture

4.3.8 STAGES OF IOT SOLUTIONS ARCHITECTURE

There are several layers of IoT built upon the capability and performance of IoT elements that provides the optimal solution to the business enterprises and end-users. The IoT architecture is a fundamental way to design the various elements of IoT, so that it can deliver services over the networks and serve the needs for the future.

Following are the primary stages (layers) of IoT that provides the solution for IoT architecture.

1. **Sensors/Actuators:** Sensors or Actuators are the devices that are able to emit, accept and process data over the network. These sensors or actuators may be connected either through wired or wireless. This contains GPS, Electrochemical, Gyroscope, RFID, etc. Most of the sensors need connectivity through sensors gateways. The connection of sensors or actuators can be through a Local Area Network (LAN) or Personal Area Network.
2. **Gateways and Data Acquisition:** As the large numbers of data are produced by this sensors and actuators need the high-speed Gateways and Networks to transfer the data. This network can be of type Local Area Network (LAN such as WiFi, Ethernet, etc.), Wide Area Network (WAN such as GSM, 5G, etc.).
3. **Edge IT:** Edge in the IoT Architecture is the hardware and software gateways that analyze and pre-process the data before transferring it to the cloud. If the data read from the sensors and gateways are not changed from its previous reading value then it does not transfer over the cloud, this saves the data used.
4. **Data center/ Cloud:** The Data Center or Cloud comes under the Management Services which process the information through analytics, management of device and security controls. Beside this security controls and device management the cloud transfer the data to the end users application such as Retail, Healthcare, Emergency, Environment, and Energy, etc.

The 4 Stage IoT Solutions Architecture

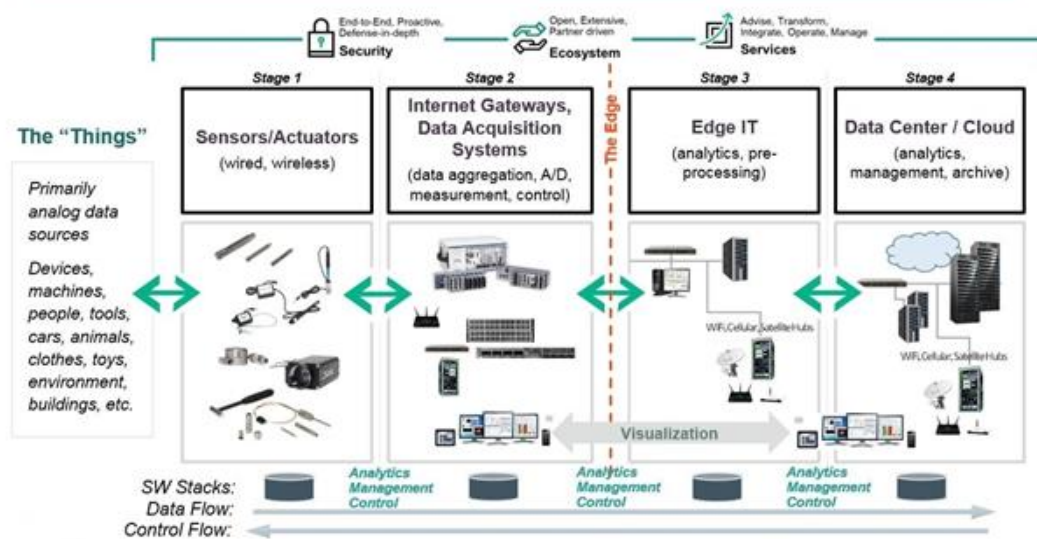


Fig 4.12

What are Smarts Objects in IoT

The concept of smart in IoT is used for physical objects that are active, digital, networked, can operate to some extent autonomously, reconfigurable and has local control of the resources. The smart objects need energy, data storage, etc.

A **smart object** is an object that enhances the interaction with other smart objects as well as with people also. The world of IoT is the network of interconnected heterogeneous objects (such as smart devices, smart objects, sensors, actuators, RFID, embedded computers, etc.) uniquely addressable and based on standard communication protocols.

In a day to day life, people have a lot of object with internet or wireless or wired connection. Such as:

- Smartphone
- Tablets
- TV computer

These objects can be interconnected among them and facilitate our daily life (smart home, smart cities) no matter the situation, localization, accessibility to a sensor, size, scenario or the risk of danger.



Fig 4.13

Smart objects are utilized widely to transform the physical environment around us to a digital world using the Internet of things (IoT) technologies.

A smart object carries blocks of application logic that make sense for their local situation and interact with human users. A smart object sense, log, and interpret the occurrence within themselves and the environment, and intercommunicate with each other and exchange information with people.

The work of smart object has focused on technical aspects (such as software infrastructure, hardware platforms, etc.) and application scenarios. Application areas range from supply-chain management and enterprise applications (home and hospital) to healthcare and industrial workplace support. As for human interface aspects of smart-object technologies are just beginning to receive attention from the environment.

4.3.9 IOT DEVICES

Internet of Things Devices is non-standard devices that connect wirelessly to a network with each other and able to transfer the data. IoT devices are enlarging the internet connectivity beyond standard devices such as smartphones, laptops, tablets, and desktops. Embedding these devices with technology enable us to communicate and interact over the networks and they can be remotely monitored and controlled.

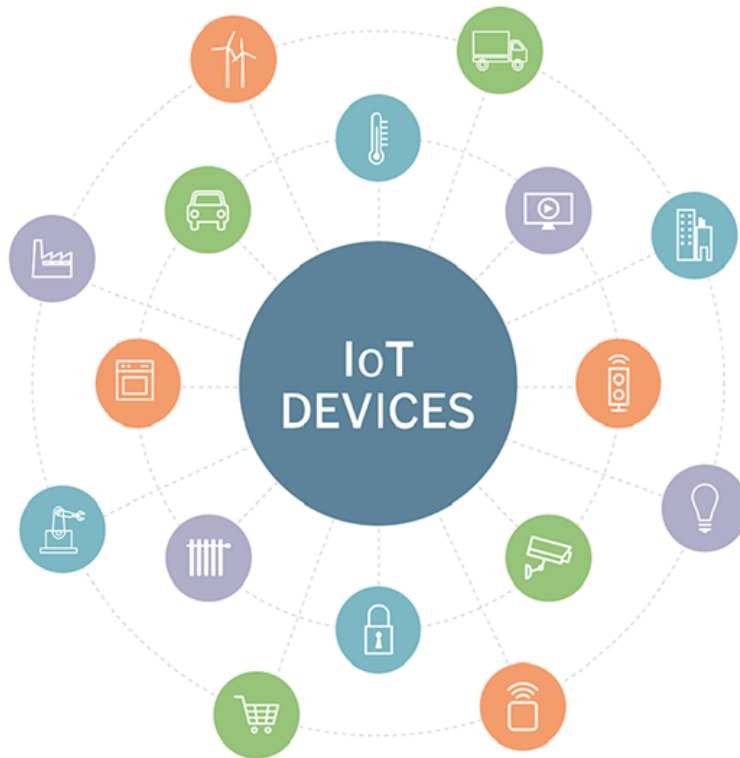
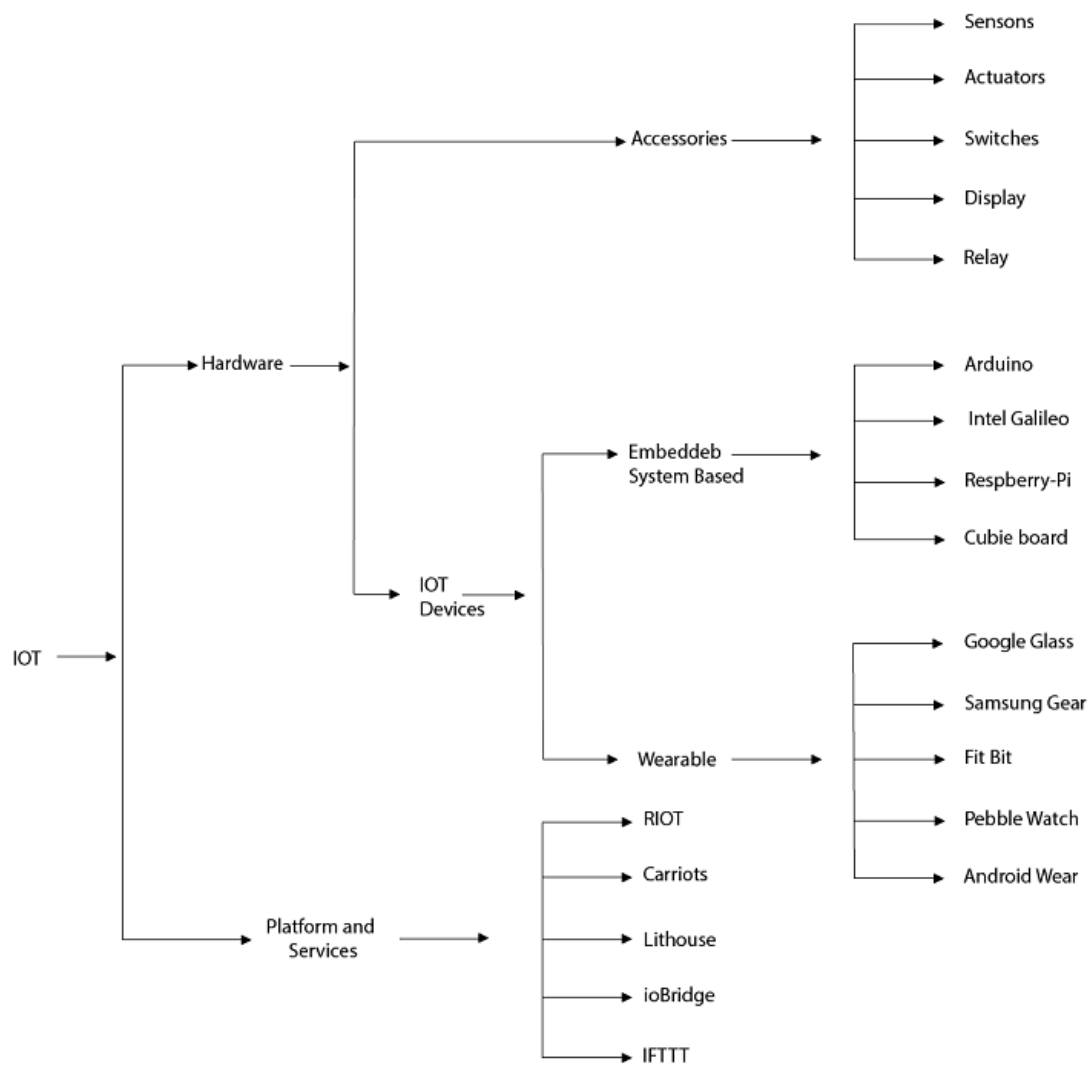


Fig 4.14 IOT Devices

There are large varieties of IoT devices available based on IEEE 802.15.4 standard. These devices range from wireless motes, attachable sensor-boards to interface-board which are useful for researchers and developers.

IoT devices include computer devices, software, wireless sensors, and actuators. These IoT devices are connected over the internet and enabling the data transfer among objects or people automatically without human intervention.

Some of the common and popular IoT devices are given below:



IoT Devices and Technologies

Fig 4.15 IOT Devices and Technologies

ARDUINO DEVICE:

Arduino devices are the microcontrollers and microcontroller kit for building digital devices that can be sense and control objects in the physical and digital world. Arduino boards are furnished with a set of digital and analog input/output pins that may be interfaced to various other circuits. Some Arduino boards include USB (Universal Serial Bus) used for loading programs from the personal computer.



Fig 4.16 Aurdino

INTEL GALILEO:

The Intel Galileo Gen 2 Board includes the parts such as Intel Quark SoC processor, 256MB RAM, multiple ports and supports for Arduino device.



Fig 4.17 Intel Galileo

SAMSUNG GEAR FIT:

A Samsung Gear Fir device is a dustproof, water-resistant with fitness tracker features, a curved display, and long-lasting battery. This device receives alerts about emails and text messages, and it integrates with Samsung's S Health app.



Fig 4.18

SENSOR:

A sensor is a device that reads the surrounding temperature, humidity, light, air quality control etc. There are different types of sensors available that reads different types of data. The sensors transmit these data over the networks or through which it is connected.



Fig 4.19

BLUETOOTH LOW ENERGY (BLE) INTELLIGENT BEACON:

A Bluetooth low energy beacon device is used to track the object located at a real time. Many companies use it to track the location of employees, assets, patients, and more in real time. This service primarily focuses on manufacturing, retail, and healthcare services.



Fig 4.20

4.3.10 PROPERTIES OF IOT DEVICES

Some of the essential properties of IoT devices are mention below:

- **Sense:** The devices that sense its surrounding environment in the form of temperature, movement, and appearance of things, etc.
- **Send and receive data:** IoT devices are able to send and receive the data over the network connection.
- **Analyze:** The devices can able to analyze the data that received from the other device over the internet networks.
- **Controlled:** IoT devices may control from some endpoint also. Otherwise, the IoT devices are themselves communicate with each other endlessly leads to the system failure.

MAJOR IOT BOARDS IN MARKET

There are several IoT boards available in the market to build the project. Some of the major IoT Boards are described below:

RASPBERRY PI:

Raspberry Pi is a much popular device used in building IoT project. The recently launched Raspberry Pi 3 includes built-in WiFi and Bluetooth making the most

compact and standalone computer. It provides a powerful environment to install a variety of programming packages such as Python, Node.js, LAMP stack, Java and much more. Using 40 GPIO pins, and four USB ports you can connect many peripherals and accessories to the Pi.



Fig 4.21

Arduino:

Arduino boards are the microcontrollers and microcontroller kit for building digital devices that can be sense and control objects in the physical and digital world. Arduino boards are furnished with a set of digital and analog input/output pins that may be interfaced to various other circuits. Some Arduino boards include USB (Universal Serial Bus) to load programs from the personal computer.



Fig 4.22

ESP8266:

The ESP8266 is a low-cost Wi-Fi microchip with 32-bit microcontroller capability, standard digital peripheral interfaces. There are different types of ESP8266 boards available for different needs. The primary goal of this board is to deal with the built-in Wifi through AT commands if used as device module, but you can 'program' using Arduino board however it also read and controls input/output, digital and analog.



Fig 4.23

SENSE HAT 8X8 RGB LED MATRIX:

Raspberry Pi Sense HAT is an integrated sensor that can measure humidity, temperature, acceleration, and pressure. The 8x8 LED matrix display data read from Raspberry Pi Sense HAT sensors. The Sense HAT has an 8×8 RGB LED matrix includes the following sensors:

1. Gyroscope
2. Accelerometer
3. Magnetometer
4. Temperature
5. Barometric pressure

6. Humidity

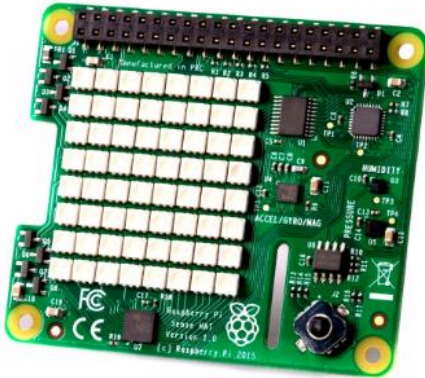


Fig 4.24

BLUETOOTH MODULE HC-05:

Bluetooth Module HC-05 device is a 6 pins Bluetooth device that is used for wireless communication. Usually, this device connects small devices like mobile phones, PDAs and TVs using a short-range wireless connection to exchange data. It communicates with the microcontroller using the serial port (USART).

PIN DESCRIPTION

- **EN:** It is the enable pin, when it is connected to 3.3V then module is enabled.
- **+5V:** This is the supply pin for connecting +5V.
- **GND:** It is the ground pin.
- **TX:** It is the transmitter pin of the UART communication.
- **RX:** It is the receiver pin of UART communication.
- **STATE:** It indicates whether the module is connected or not. It acts as a status indicator.



Fig 4.25

IOT - PLATFORM

As in IoT, all the IoT devices are connected to other IoT devices and application to transmit and receive information using protocols. There is a gap between the IoT device and IoT application. An IoT Platform fills the gap between the devices (sensors) and application (network). Thus we can say that **an IoT platform is an integrated service that fulfills the gap between the IoT device and application and offers you to bring physical object online.**



Fig 4.26

There are several IoT Platforms available that provides facility to deploy IoT application actively. Some of them are listed below:

Amazon Web Services (AWS) IoT platform: Amazon Web Service IoT platform offers a set of services that connect to several devices and maintain the security as well. This platform collects data from connected devices and performs real-time actions.

Microsoft Azure IoT platform: Microsoft Azure IoT platform offers strong security mechanism, scalability and easy integration with systems. It uses standard protocols that support bi-directional communication between connected devices and platform. Azure IoT platform has an Azure Stream Analytics that processes a large amount of information in real-time generated by sensors. Some common features provided by this platform are:

- Information monitoring

- A rules engine
- Device shadowing
- Identity registry

Google Cloud Platform IoT: Google Cloud Platform is a global cloud platform that provides a solution for IoT devices and applications. It handles a large amount of data using Cloud IoT Core by connecting various devices. It allows to apply BigQuery analysis or to apply Machine learning on this data. Some of the features provided by Google Cloud IoT Platform are:

- Cloud IoT Core
- Speed up IoT devices
- Cloud publisher-subscriber
- Cloud Machine Learning Engine

IBM Watson IoT platform: The IBM Watson IoT platform enables the developer to deploy the application and building IoT solutions quickly. This platform provides the following services:

- Real-time data exchange
- Device management
- Secure Communication
- Data sensor and weather data services

Artik Cloud IoT platform: Arthik cloud IoT platform is developed by Samsung to enable devices to connect to cloud services. It has a set of services that continuously connect devices to the cloud and start gathering data. It stores the incoming data from connected devices and combines this information. This platform contains a set of connectors that connect to third-party services.

Bosch IoT Suite:

Bosch cloud IoT Suit is based on Germany. It offers safe and reliable storing of data on its server in Germany. This platform supports full app development from prototype to application development.

How IoT platform help:

- IoT Platform connects sensors and devices.
- IoT platform handles different software communication protocol and hardware.
- IoT platform provides security and authentication for sensors and users.
- It collects, visualizes, and analyzes the data gathered by the sensor and device.

4.4 RASPBERRY PI

4.4.1 Raspberry Pi Basic Hardware Setup

Extra Hardware

You Will Need The Raspberry Pi board contains a processor and graphics chip, program memory (RAM) and various interfaces and connectors for external devices. Some of these devices are essential, others are optional. RPi operates in the same way as a standard PC, requiring a keyboard for command entry, a display unit and a power supply. It also requires 'mass-storage', but a hard disk drive of the type found in a typical PC is not really in keeping with the miniature size of RPi. Instead we will use an SD Flash memory card normally used in digital cameras, configured in such a way to 'look like' a hard drive to RPi's processor. RPi will 'boot' (load the Operating System into RAM) from this card in the same way as a PC 'boots up' into Windows from its hard disk.

The following are essential to get started:

- SD card containing Linux Operating system

- USB keyboard
- TV or monitor (with HDMI, DVI, Composite or SCART input)
- Power supply (see Section 1.6 below)
- Video cable to suit the TV or monitor used recommended optional extras include:
 - USB mouse
 - Internet connection, Model A or B: USB WiFi adaptor
 - Internet connection, Model B only: LAN (Ethernet) cable
 - Powered USB hub

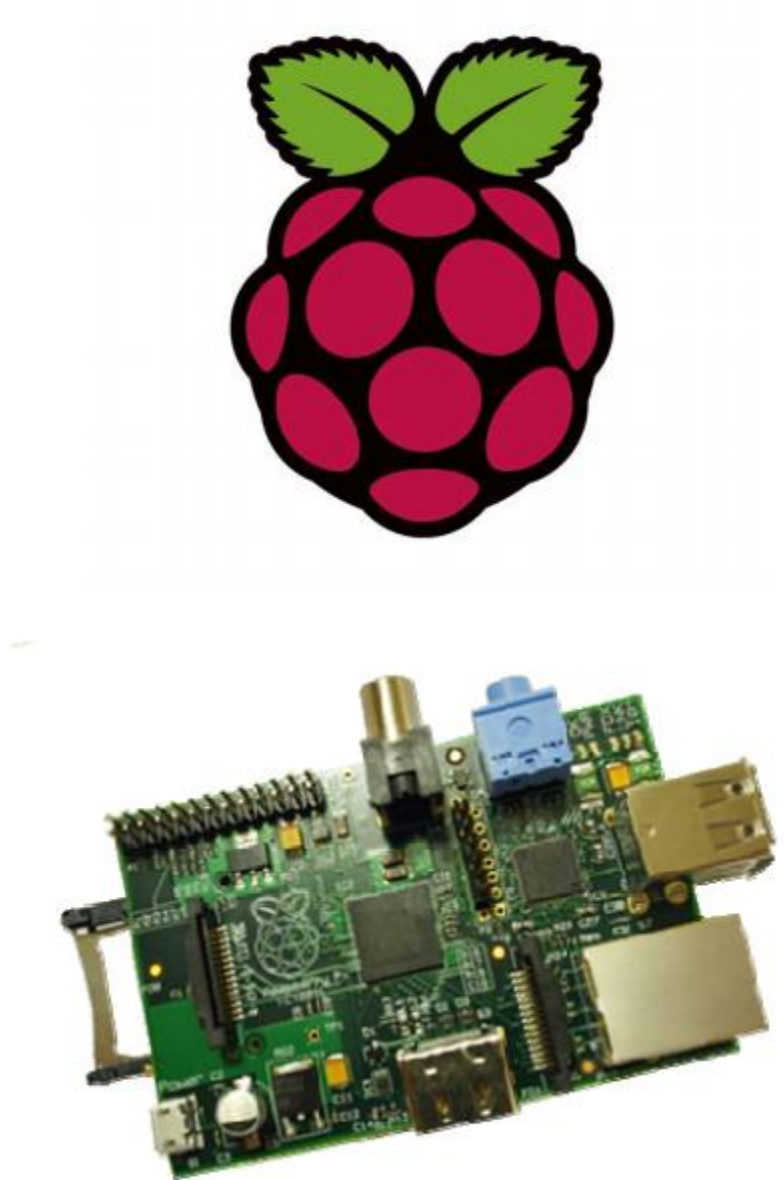


Fig 4.27 Connecting Everything Together

1. Plug the preloaded SD Card into the RPi.
2. Plug the USB keyboard and mouse into the RPi, perhaps via a USB hub. Connect the Hub to power, if necessary.
3. Plug a video cable into the screen (TV or monitor) and into the RPi.
4. Plug your extras into the RPi (USB WiFi, Ethernet cable, external hard drive etc.). This is where you may really need a USB hub.

5. Ensure that your USB hub (if any) and screen are working.
6. Plug the power supply into the mains socket.
7. With your screen on, plug the power supply into the RPi microUSB socket.
8. The RPi should boot up and display messages on the screen. It is always recommended to connect the MicroUSB power to the unit last (while most connections can be made live, it is best practice to connect items such as displays with the power turned off). The RPi may take a long time to boot when powered-on for the first time, so be patient!

4.4.2 OPERATING SYSTEM SD CARD:

As the RPi has no internal mass storage or built-in operating system it requires an SD card preloaded with a version of the Linux Operating System. • You can create your own preloaded card using any suitable SD card (4GBytes or above) you have to hand. We suggest you use a new blank card to avoid arguments over lost pictures. • Preloaded SD cards will be available from the RPi Shop.

4.4.3 KEYBOARD & MOUSE:

Most standard USB keyboards and mice will work with the RPi. Wireless keyboard/mice should also function, and only require a single USB port for an RF dongle. In order to use a Bluetooth keyboard or mouse you will need a Bluetooth USB dongle, which again uses a single port. Remember that the Model A has a single USB port and the Model B has two (typically a keyboard and mouse will use a USB port each).

4.4.4 DISPLAY:

There are two main connection options for the RPi display, HDMI (High Definition) and Composite (Standard Definition). • HD TVs and many LCD monitors can be connected using a full-size 'male' HDMI cable, and with an inexpensive adaptor if DVI is used. HDMI versions 1.3 and 1.4 are supported and a version 1.4 cable is recommended. The RPi outputs audio and video via HDMI, but does not support HDMI input. • Older TVs can be connected using Composite video (a yellow-to-yellow RCA cable) or via SCART (using a Composite video to SCART adaptor). Both PAL and NTSC format TVs are supported. When using a composite

video connection, audio is available from the 3.5mm jack socket, and can be sent to your TV, headphones or an amplifier. To send audio to your TV, you will need a cable which adapts from 3.5mm to double (red and white) RCA connectors. Note: There is no analogue VGA output available. This is the connection required by many computer monitors, apart from the latest ones. If you have a monitor with only a D-shaped plug containing 15 pins, then it is unsuitable.

4.4.5 RASPBERRY PI PORT SPECIFICATION:

The following diagram shows the port specification

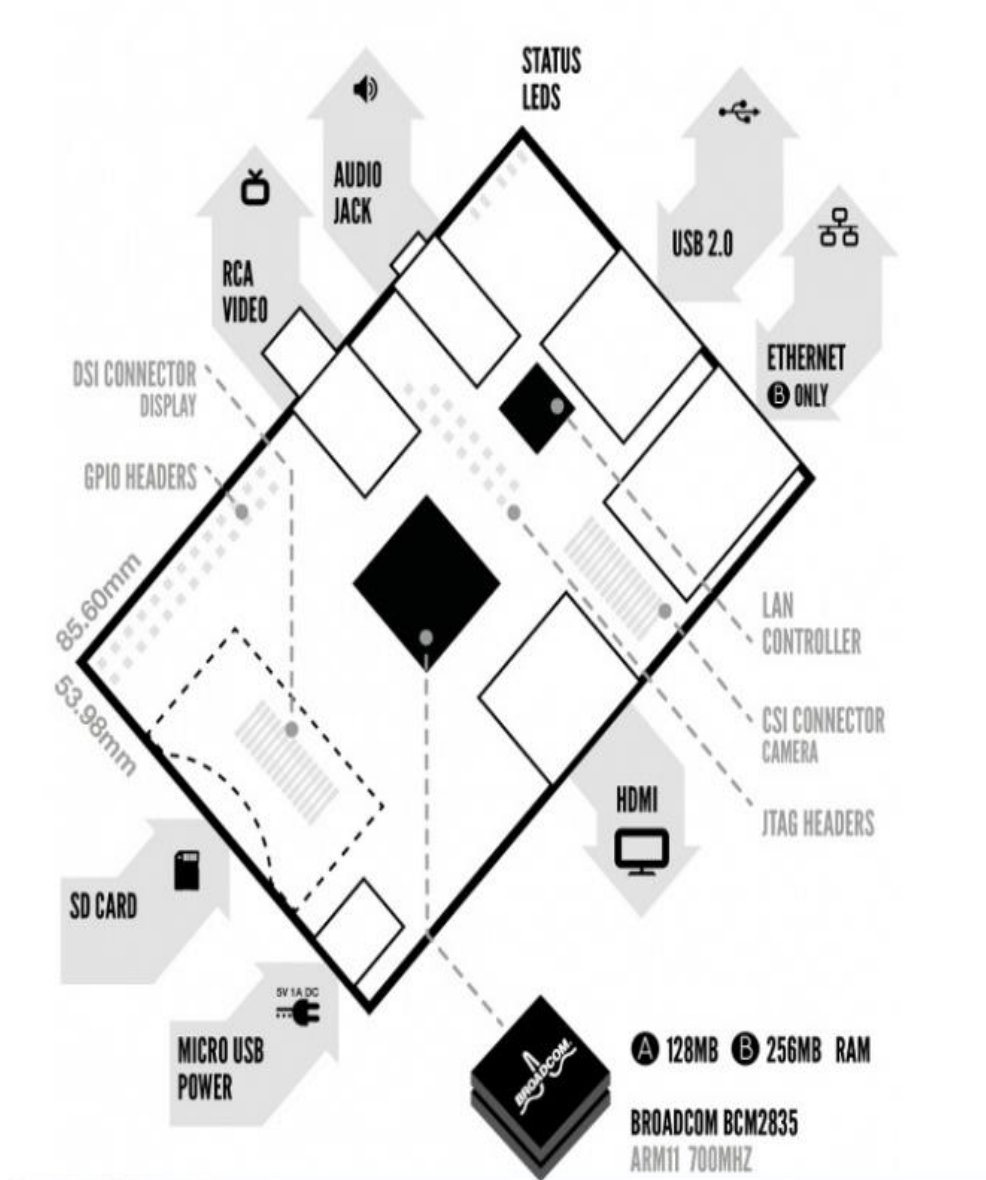


Fig 4.28 Port Specification

4.4.6 POWER SUPPLY:

The unit is powered via the microUSB connector (only the power pins are connected, so it will not transfer data over this connection). A standard modern phone charger with a microUSB connector will do, providing it can supply at least 700mA at +5Vdc. Check your power supply's ratings carefully. Suitable mains adaptors will be available from the RPi Shop and are recommended if you are unsure what to use.

Note: The individual USB ports on a powered hub or a PC are usually rated to provide 500mA maximum. If you wish to use either of these as a power source then you will need a special cable which plugs into two ports providing a combined current capability of 1000mA.

4.4.7 CABLES:

You will need one or more cables to connect up your RPi system.

- Video cable alternatives: o HDMI-A cable o HDMI-A cable + DVI adapter o Composite video cable o Composite video cable + SCART adaptor
- Audio cable (not needed if you use the HDMI video connection to a TV)
- Ethernet/LAN cable (Model B only)

4.4.8 ADDITIONAL PERIPHERALS:

You may decide you want to use various other devices with your RPi, such as Flash Drives/Portable Hard Drives, Speakers etc.

Internet Connectivity:

This may be via an Ethernet/LAN cable (standard RJ45 connector) or a USB WiFi adaptor. The RPi Model B Ethernet port is auto-sensing which means that it may be connected to a router or directly to another computer (without the need for a crossover cable).

USB hub:

In order to connect additional devices to the RPi, you may want to obtain a USB hub, which will allow multiple devices to be used. It is recommended that a powered hub is used - this will provide any additional power to the devices without affecting the RPi itself. A USB 2.0 model is recommended. USB 1.1 is fine for keyboards and mice, but may not be fast enough for other accessories.

Case:

Since the RPi is supplied without a case, it will be important to ensure that you do not use it in places where it will come into contact with conductive metal or liquids, unless suitably protected.

Expansion & Low-Level Peripherals:

If you plan on making use of the low-level interfaces available on the RPi, then ensure you have a suitable plug for the GPIO header pins. Also if you have a particular low-level project in mind, then ensure you design-in suitable protection circuits to keep your RPi safe.

4.5 RASPBERRY PI ADVANCED SETUP (GEEKS ONLY)

4.5.1 Finding hardware and setting up:

You'll need a preloaded SD card, USB keyboard, TV/Monitor (with HDMI/ DVI/ Composite/SCART input), and power supply (USB charger or a USB port from a powered USB Hub or another computer). You'll likely also want a USB mouse, a case, and a USB hub (a necessity for Model A). A powered USB hub will reduce the demand on the RPi. To connect to the Internet, you'll need either an Ethernet/LAN cable (Model B) or a USB WiFi adaptor (either model). When setting up, it is advisable to connect the power after everything else is ready.

4.5.2. SERIAL CONNECTION:

The Serial Port is a simple and uncomplicated method to connect to the RPi. The communication depends on byte wise data transmission, is easy to setup and is generally available even before boot time.

First interaction with the board:

Connect the serial cable to the COM port in the RPi, and connect the other end to the COM port or USB Serial Adapter in the computer.

4.5.3 SERIAL PARAMETERS:

The following parameters are needed to connect to the RPi. All parameters except Port_Name and Speed are default values and may not need to be set. Port_Name: Linux automatically assigns different names for different types of serial connectors.

Choose your option:

- Standard Serial Port: ttyS0 ... ttySn

- USB Serial Port Adapter: ttyUSB0 ... ttyUSBn
- Speed: 115200
- Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None The Serial Port is generally usable by the users in the group dialout. To add oneself to the group dialout the the following command needs to be executed with root privileges:

```
$useradd -G {dialout} your_name
```

Super Easy Way using GNU Screen:

Enter the command below into a terminal window

```
screen Port_Name 115200
```

Super Easy Way using Minicom Run minicom with the following parameters:

```
minicom -b 115200 -o -D Port_Name
```

GUI method with GtkTerm:

Start GtkTerm,

select Configuration->Port and enter the values above in the labelled fields.

Windows Users:

Windows 7 or Vista users must download putty or a comparable terminal program. Users of XP and below can choose between using putty and HyperTerminal.

First Dialog:

If you get the prompt below, you are connected to the Raspberry Pi shell!

```
prompt> #
```

First command you might want try is "help":

```
prompt> # help
```

If you get some output, you are correctly connected to the Raspberry Pi! Congratulations!

4.6 SD CARD SETUP:

Now we want to install a GNU/Linux distro on an SD card and make space for our stuff. You can use either an SD or SDHC card. In the latter case of course take care that your PC card reader also supports SDHC. Be aware that you are not dealing with an x86 processor, but instead a completely different architecture called ARM, so don't forget to install the ARM port for the distro you are planning to use.

Formatting the SD card via the mkcard.txt script:

1. Download mkcard.txt .
2. \$ chmod +x mkcard.txt
3. \$./mkcard.txt /dev/sdx, where x is the letter of the card.

You can find this by inserting your card and then running `dmesg | tail`. You should see the messages about the device being mounted in the log. Mine mounts as sdc. Once run, your card should be formatted.

4.7 LCD DISPLAY

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal.

An LCD consists of two glass panels, with the liquid crystal material sandwiched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed. Polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle.

One each polarisers are pasted outside the two glass panels. These polarisers would rotate the light rays passing through them to a definite angle, in a particular direction

When the LCD is in the off state, light rays are rotated by the two polarisers and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarisers, which would result in activating / highlighting the desired characters.

The LCD's are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations.

The LCD's don't generate light and so light is needed to read the display. By using backlighting, reading is possible in the dark. The LCD's have long life and a wide operating temperature range.

Changing the display size or the layout size is relatively simple which makes the LCD's more customer friendly.

The LCDs used exclusively in watches, calculators and measuring instruments are the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCDs being extensively used in telecommunications and entertainment electronics. The LCDs have even started replacing the cathode ray tubes (CRTs) used for the display of text and graphics, and also in small TV applications.

Crystalonics dot –matrix (alphanumeric) liquid crystal displays are available in TN, STN types, with or without backlight. The use of C-MOS LCD controller and driver ICs result in low power consumption. These modules can be interfaced with a 4-bit or 8-bit micro processor /Micro controller.

The built-in controller IC has the following features:

- Correspond to high speed MPU interface (2MHz)
- 80 x 8 bit display RAM (80 Characters max)
- 9,920 bit character generator ROM for a total of 240 character fonts. 208 character fonts (5 x 8 dots) 32 character fonts (5 x 10 dots)
- 64 x 8 bit character generator RAM 8 character generator RAM 8 character fonts (5 x 8 dots) 4 characters fonts (5 x 10 dots)
- Programmable duty cycles
 - 1/8 – for one line of 5 x 8 dots with cursor
 - 1/11 – for one line of 5 x 10 dots with cursor
 - 1/16 – for one line of 5 x 8 dots with cursor
- Wide range of instruction functions display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift.

4.8 FUNCTIONAL DESCRIPTION OF THE CONTROLLER IC

REGISTERS:

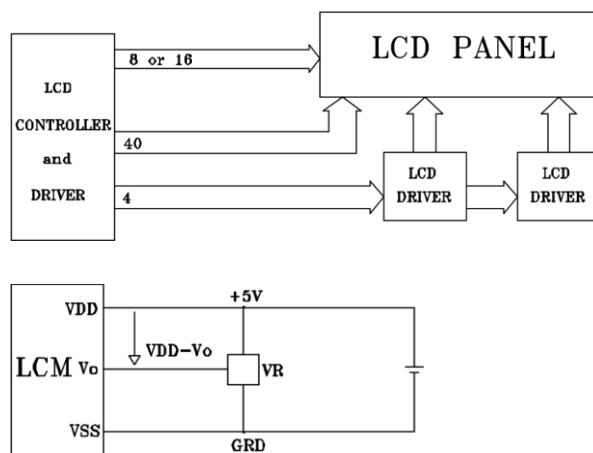
The controller IC has two 8 bit registers, an instruction register (IR) and a data register (DR). The IR stores the instruction codes and address information for display data RAM (DD RAM) and character generator RAM (CG RAM). The IR can be written, but not read by the MPU.

The DR temporally stores data to be written to /read from the DD RAM or CG RAM. The data written to DR by the MPU, is automatically written to the DD RAM or CG RAM as an internal operation.

When an address code is written to IR, the data is automatically transferred from the DD RAM or CG RAM to the DR. data transfer between the MPU is then completed when the MPU reads the DR. likewise, for the next MPU read of the DR, data in DD RAM or CG RAM at the address is sent to the DR automatically. Similarly, for the MPU write of the DR, the next DD RAM or CG RAM address is selected for the write operation.

The dot-matrix liquid crystal display controller and driver LSI displays alphanumeric, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

Block Diagram



VDD-Vo: LCD DRIVING VOLTAGE

VR: 10K-20K.

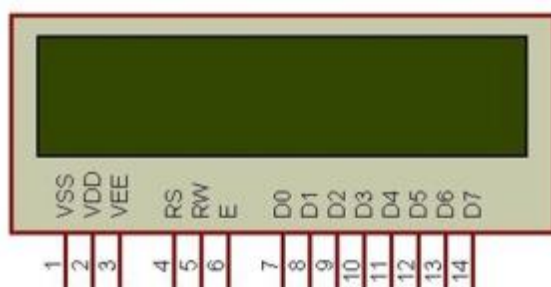


Fig 4.29

Absolute Maximum Ratings

Item	Symbol	Min	Max	Unit
Power Voltage	VDD - VSS	0	7.0	V
Input Voltage	V _{in}	V _{SS}	VDD	
Operating Temperature Range	T _{OP}	0	+50	T
Storage Temperature Range	T _{ST}	-20	+60	

4.9 Introduction: Make Your Raspberry Pi Speak!



Fig 4.30

Sometimes your robot just wants to talk! This tutorial will show you how to make your Raspberry Pi speak aloud. We will use a software package called Espeak to

convert text (or strings) into spoken words, out loud, on your robot. We show you three examples:

- [How to create a countdown.](#)
- [How to modify the voice.](#)
- Modify the voice for speed, language, and gender.

Video

Here's a quick video that summarizes the information below and walks you through step by step on getting the Raspberry Pi to speak aloud.

Materials You'll Need

- A [Raspberry Pi](#)
- [Raspbian for Robots SD Card](#)
- [Power Supply](#)
- [Wifi Dongle](#)



Fig 4.31

Dexter Industries Speakers

All the instructions assume you're using [Raspbian for Robots](#), our custom operating system for the Raspberry Pi based on Raspbian. You can buy an SD

Card with Raspbian for Robots on it [here](#), [or you can download it for free and install it using our directions here](#).

Software

This tutorial uses **python3**. It can be adapted to earlier versions of python.

Connections

- Make sure your Raspberry Pi is powered up and connected to your network.
- Connect the speaker to the audio jack as shown in the image.

```
libson: ~$ sudo apt-get install espeak
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  espeak-data libespeak1 libsonnet0
The following NEW packages will be installed:
  espeak espeak-data libespeak1 libsonnet0
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,182 kB of archives.
After this operation, 2,556 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://mirrorsirector.raspbian.org/raspbian/ jessie/main libsonnet0 armhf 0.1.17-1.1 [112.2 kB]
Get:2 http://mirrorsdirector.raspbian.org/raspbian/ jessie/main espeak-data armhf 1.48.04+dfsg-1 [941 kB]
Get:3 http://mirrorsdirector.raspbian.org/raspbian/ jessie/main libespeak1 armhf 1.48.04+dfsg-1 [139 kB]
Get:4 http://mirrorsdirector.raspbian.org/raspbian/ jessie/main espeak armhf 1.48.04+dfsg-1 [70.9 kB]
Fetched 1,182 kB in 1s (79.1 kB/s)
Selecting previously unselected package libsonnet0:armhf.
(Reading database ... 12455 files and directories currently installed.)
Preparing to unpack .../libsonnet0_0.1.17-1.1_armhf.deb ...
Unpacking libsonnet0:armhf (0.1.17-1.1) ...
Selecting previously unselected package espeak-data:armhf.
Preparing to unpack .../espeak-data_1.48.04+dfsg-1_armhf.deb ...
Unpacking espeak-data:armhf (1.48.04+dfsg-1) ...
Selecting previously unselected package libespeak1:armhf.
Preparing to unpack .../libespeak1_1.48.04+dfsg-1_armhf.deb ...
Unpacking libespeak1:armhf (1.48.04+dfsg-1) ...
Selecting previously unselected package espeak.
Preparing to unpack .../espeak_1.48.04+dfsg-1_armhf.deb ...
Unpacking espeak (1.48.04+dfsg-1) ...
Processing triggers for man-db (2.7.5-2-5) ...
Setting up libsonnet0:armhf (0.1.17-1.1) ...
Setting up espeak-data:armhf (1.48.04+dfsg-1) ...
Setting up libespeak1:armhf (1.48.04+dfsg-1) ...
Setting up espeak (1.48.04+dfsg-1) ...
Processing triggers for libc-bin (2.19-18-debdeb4) ...
```

Fig 4.32

Setting up the TTS (Text To Speech) Engine

To make the Raspberry Pi speak and read some text aloud, we need a software interface to convert text to speech on the speakers. For this we need a **Text To Speech engine**. The TTS engine we are using in this tutorial is [eSpeak](#). The voice may be a little robotic, however it runs offline which is an added plus.

First let's test if the audio is working on the Raspberry Pi. Run the following command:

```
aplay /usr/share/sounds/alsa/*
```

If you are able to hear the sounds like “**Front Center**”, “**Front Left**”, “**Front Right**” and so on, your sound is working!

Next, install eSpeak. Run the following in terminal to install espeak:

```
sudo apt-get install espeak
```

Get the Raspberry Pi Speaking from the Command Line

After eSpeak has been successfully installed on the Raspberry Pi, run the following command to test eSpeak:

```
espeak "Text you wish to hear back" 2>/dev/null
```

For example:

```
espeak "Hello World" 2>/dev/null
```

You should hear “**Hello World**” from the speakers. Your Raspberry Pi is speaking!

Using Python to Make Your Raspberry Pi Speak

Now building on top of these commands you can create Python programs that speak words. [We have two examples that you can download here](#):

- [speak_count.py](#) – This program reads aloud a countdown. The program prompts you for a number, and then counts off the numbers out loud until it reaches 0.
- [speak_text.py](#) – This program reads aloud the text you enter and it also records the speech to a file named Text.wav.

Running the Example Programs:

Counting Numbers

Connect your Pi to the internet. You will need to download and install the *num2words* Python package to convert the numbers to strings. This will convert an integer value to a string, which are then read aloud by the Raspberry Pi. Use the following command:

```
sudo pip3 install num2words
```

Go to the folder where you have copied the examples and run the following command:

```
sudo python3 speak_count.py
```

Enter a number and you should hear a voice saying “**Count Down Starts**”, then you should hear numbers counted down from the entered number.

```
pi@dex:~ $ sudo python3 speak_count.py
Enter a number: 5
five Count Down Starts
1
2
3
4
5
```

Fig 4.33

Speaking Commands

Run the program “speak_text.py” in the command line:

```
sudo python3 speak_text.py
```

When prompted, enter some text you wish to hear back. You should see the text you have entered and you will hear it back through your speakers!

Sample output is shown below.

```
pi@dex:~ $ sudo python3 speak_text.py
Enter the Text: Twinkle Twinkle Little Star
Twinkle Twinkle Little Star
Playing WAVE '/home/pi/Desktop/Text.wav' : Signed 16 bit Little Endian, Rate 22050 Hz
```

Fig 4.34

As a bonus, this example also records the voice to a file named **Text.wav**.

Run the following commands to hear back the recorded voice.

```
aplay /home/pi/Desktop/Text.wav
```

A Deeper Dive on Raspberry Pi Speech: More Options

Modifying the Voice

The Espeak package provides a few good variations to the default voice used to speak out the text. The software supports [more than 30 languages](#) and a few of them are shown below. The syntax is “-v<voice filename>” For example, we can use an American accent with “en-us”:

```
espeak -ven-us "Welcome to Dexter tutorial" 2>/dev/null
```

Or we can use a French accent with **“fr”**:

```
espeak -vfr "Ce tutoriel de Dexter vous souhaite la bienvenue" 2>/dev/null
```

Or we can speak Spanish with the Raspberry Pi using the term **“es”**:

```
espeak -ves "Bienvenido al tutorial de Dexter" 2>/dev/null
```

Voices

You can also specify a male voice, or a female voice. You can choose between a few different male voices :**+m1,+m2,+m3,+m4,+m5,+m6,+m7**. For example, we can speak in male voice five:

```
espeak -ven+m5 "Welcome to Dexter tutorial" 2>/dev/null
```

You can also use female voices: **+f1,+f2,+f3,+f4**. For example, we can speak in female voice four:

```
espeak -ven+f4 "Welcome to Dexter tutorial" 2>/dev/null
```

You can also modify a few other aspects of the voice with **+croak**, **+whisper**. For example, we can whisper:

```
espeak -ven+whisper "Welcome to Dexter tutorial" 2>/dev/null
```

Modifying Speech Speed

The **“-s”** parameter controls the speed of reading a single word. It sets the speed in words-per-minute. The default value is 175. I generally use a faster speed of 260. The lower limit is 80. There is no upper limit, but about 500 is probably a practical maximum. We can set the word speed like this:

Fast at 300 words per minute:

```
espeak -s250 "Welcome to Dexter tutorial" 2>/dev/null
```

Or slow at 80 words per minute:

```
espeak -s80 "Welcome to Dexter tutorial" 2>/dev/null
```

Pause Between Words

We can use the “-g” to pause between words; the “g” stands for “word gap”. This option inserts a pause between words. The value is the length of the pause, in units of 10 ms (at the default speed of 170 wpm). A fast pause:

```
espeak -g10 "Welcome to Dexter tutorial" 2>/dev/null
```

A slow pause with 100 ms between words:

```
espeak -g180 "Welcome to Dexter tutorial" 2>/dev/null
```

There are lot more options available with the eSpeak package you can review [here](#).

CHAPTER 5

5.1 DISCUSSION

The Following figures shows the hardware set up of the project and the pictures from the monitor shows the scheduled time and the following medicine that should be taken by the patient.

5.2 HARDWARE KIT

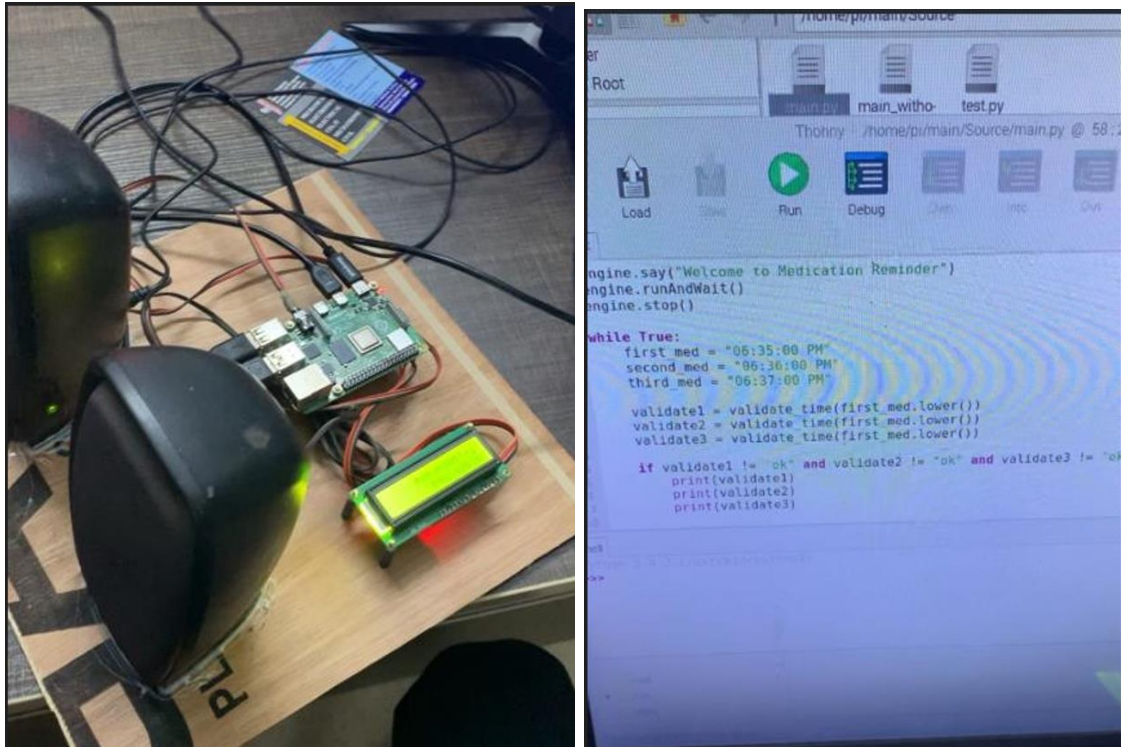


Fig 5.1



Fig 5.2

5.3 Result

The proposed system Can clearly save human beings from extreme effect of improper medication mix-ups and remaind the patients by a voice to take the correct medicine from time to time. It can be easily handled by illiterate users to store inputs according to their convenience.

CHAPTER 6

6.1 CONCLUSIONS

Many Medication Reminder Systems have been developed on different platforms. Many of these systems require special hardware devices to remind the patients about the medicine in-take timings.

Purchasing new hardware devices becomes costly and more time and money consuming. So in the given work an attempt has been made to implement a system which is economical, easily accessible and improves medication adherence. Medication non-adherence reduces the effectiveness of a treatment and imposes a financial burden on health care systems.

The patients will get the schedule of medicine in-take time with medicine description, starting and ending date of medicine, notification through message or email, automatic alarm ringing system and navigation system.

The scheduled reminder will not suggest any kind of medicine which is not prescribed by the doctor that will assure the safety of the patient and also will avoid wrong dosages.

The patients can also search doctors disease wise (depending upon the specialization of the doctor), which provides easy searching facility to the users and saves the time. Doctors can view all the fixed appointments along with date and time, which he fixed and through this he can make new appointment schedules.

6.2 References:

- [1] Ammouri, S. and Bilodeau, G.A. (2008) "Face and hands detection and tracking applied to the monitoring of medication intake", Proc. of Canadian Conf. on Computer and Robot Vision, May, pp. 147-154.
- [2] Batz, D., Batz, M., Lobo, N.D.V. and Shah, M. (2005) "A computer vision system for monitoring medication intake", Canadian Conf. on Computer and Robot Vision, May, pp. 362-369.
- [3] Becker, E., Metsis, V., Arora, R., Vinjumur, J.K., Xu, Y. and Makedon, F. (2009) "Smart Drawer: RFID- Based smart medicine drawer for assistive environments", Proc. of Pervasive technologies related to assistive environments, June, pp 1-8.
- [4] Prasad, B., (2013) "Social media, health care, and social networking", Gastrointest Endosc. Vol. 77, pp 492–495.
- [5] Slagle, J.M., Gordon, J.S., Harris, C.E., Davison, C.L., Culpepper, D.K., Scott P. and Johnson, K.B., (2011) "MyMediHealth – Designing a next generation system for child-centered medication management", Journal of Biomedical Informatics, Vol. 43, No. 5, pp. 27-31.
- [6] Smart Medication Dispenser Suraj Shinde, Nitin Bange, Monika Kumbhar, Snehal Patil Assistant Professor, SETI, Panhala1 UG Student, SETI, Panhala International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume6, Issue4, April2017
- [7] World Population Ageing: 1950-2050, United Nations Population Division.